OIPE JC46
MAY 2 0 2004
PATENT & TRADEMARK OFFICE

#28
Appeal
PATENT
5/23/04

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Edwards	Examiner: D. Chung
Serial No.:	09/353,887	Group Art Unit: 2672
Filed:	July 15, 1999	Docket: 18195.29
Title:	GRAPHICS PROCESSOR WITH TEXTURE MEMORY ALLOCATION SYSTEM	

APPEAL BRIEF

RECEIVED
MAY 2 4 2004
Technology Center 2600

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir,

Pursuant to 37 C.F.R. 1.192, Applicant submits the following Appeal Brief:

REAL PARTY IN INTEREST

The real party in interest in the present application and Appeal is 3Dlabs Inc. Ltd., a corporation of Bermuda, having a place of business at Huntsville, Alabama 35824. 3Dlabs is a subsidiary of Creative Technologies, Ltd., a corporation of Singapore.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

STATUS OF CLAIMS

Claims 1, 4-13, 15-19, 21-22, 24-25, and 35-38 were rejected under 35 U.S.C. §103(a) as being obvious over Lentz (U.S. Pat. No. 5,886,705) in view of Young, et al (U.S. Pat. No. 5,831,637) and Tanaka, et al. (U.S. Pat. No.5,793,376), and further in view of Saunders, et al (U.S. Pat. No. 6,046,747) and Kobayashi, et al. (U.S. Pat. No. 5,761,401), and claims 14, 20, and 26-34 were rejected under 35 U.S.C. §103(a) as being obvious over Lentz, Young, et al. and

05/21/2004 AWONDAF1 00000097 09353887
01 FC:1402 330.00 OP

Tanaka, et al. in view of Saunders, et al., and further in view of Chimoto (U.S. Pat. No. 5,550,961).

STATUS OF AMENDMENTS

No amendment has been filed since the Final Rejection of the last final Office Action.

SUMMARY OF INVENTION

In summary, the invention is a graphics accelerator for processing graphical images. The graphics accelerator includes a single texture buffer and a plurality of texture processors (page 5, line 29 – page 6, line 20). Each of texture processor retrieves texture packets from the single texture buffer (page 7, lines 1-3) and each texture map associates with a texture map that is different from texture maps associated with other texture packets (page 8, lines 20-23). Each texture packet includes data relating to its associated texture map in the texture buffer (page 8, lines 16-20).

ISSUES

The following issues are presented for the Board's consideration:

1. May claims 1, 4-8, 21-22, and 24-25 be rejected under 35 U.S.C. §103(a) based upon combination of five (5) references when none of the references specifically suggest the combination?
2. May claims 26-28, 29-31, and 32-38 be rejected under 35 U.S.C. §103(a) based upon combination of four (4) references when none of the references specifically suggest the combination?
3. May claims 9-14 and 15-20 be rejected under 35 U.S.C. §103(a) based upon combination of three (3) references when none of the references specifically suggest the combination?

GROUPING OF CLAIMS

1. Claims 1, 4-8, 21-22, and 24-25 are grouped together, where the base claims, claim 1 and claim 21, are rejected by a combination of five (5) references.

2. Claims 26-28, 29-31, and 32-38 are grouped together, where the base claims, claim 26, claim 29, and 32, are rejected by a combination of four (4) references.
3. Claims 9-14 and 15-20 are grouped together, where the base claims, claim 9 and claim 15, are rejected by a combination of three (3) references.

ARGUMENT

Claim Rejections Under 35 U.S.C. §103

A. SUMMARY OF ARGUMENTS.

This section summarizes Applicant's arguments, according to the format of 37 CFR § 1.192 (c)(8)(iv). A more detailed argument and citation to authority is found below.

1. A valid teaching, suggestion, or motivation to combine the references as suggested has not been shown.

The cited references do not suggest or motivate their combination to provide functions of the invention.

2. The present claimed invention cannot be used as the motivation for combining the references.

The examiner cannot use the invention as a road map to pick and choose references to form a combination that provides all elements of the invention.

B. DETAILED ARGUMENTS AND CITATIONS TO AUTHORITY.

1. References used for rejection under 35 U.S.C. §103 must provide some suggestion for combination.

MPEP 706.02(j) states that three basic criteria must be met to establish a prima facie case of obviousness under 35 U.S.C. §103. First, there must be some suggestion or motivation, either in the references or in the knowledge generally available to one of ordinary skill in the art to combine reference teachings. Second, there must be a reasonable expectation of success. Third, the references must teach or suggest all the claim limitations.

United States Court of Appeals for Federal Circuit has also stated that "when a rejection depends on a combination of prior art references, there must be some teaching, suggestion, or motivation to combine the references." *In Re Rouffet*, 149 F. 3d 1350, 1355 (CAFC 1998). The

court further said that when combining known elements, the question is "whether there is something *in the prior art as a whole* to suggest the desirability, and thus the obviousness, of making the combination." *Id.* at 1356. (emphasis added)

In rejecting the claims, the Examiner cited six patents and one on-line dictionary, wherein the cited patents covering technologies ranging from texture memory organization to video stream data mixing and to parallel image generation. In particular, when rejecting claim 1, the Examiner admitted that Lentz (U.S. Pat. No. 5,886,705) does not teach all elements of claim 1 (Office Action of Jan. 12, 2004 (OA), page 3, ¶2). The Examiner then relied an on-line computer dictionary to interpret the buffer (OA, page 3, ¶2). The Examiner further cited Young, et al. (U.S. Pat. No. 5,831,637) for teaching of multiple texture processors and stated that the minimization of processing time as the motivation to combine Young, et al. with Lent and the on-line computer dictionary (OA, page 4, lines 1-4). The Examiner admitted that Young, et al. does not teach one single texture memory but it said that the plurality of processors retrieving data from a single texture buffer would be obvious to reduce required processing time in parallel structure (OA, page 4, lines 14-20). The Examiner cited Kobayashi, et al. (U.S. Pat. No. 5,761,401) to support this statement (OA, page 4, line 21).

The Examiner continued by admitting that Lentz does not teach texture packets identifying the location of texture maps (OA, page 5, ¶2) and cited Tanaka, et al. (U.S. Pat. No. 5,793,376) as disclosing packet data that represent the storage location of texture data/map. The Examiner did not give any reason or motivation for combing Tanaka, et al. with Lentz, the on-line computer dictionary, Young et al, and Kobayashi, et al.

The Examiner then stated that the above combination does not teach texture packets having data related to a dimensional type of its texture map (OA, page 5, ¶3). The Examiner then cited Saunders et al (U.S. Pat. No. 6,046,747) as disclosing a special bin texture call that includes a target parameter defining the dimension of the texture map (OA, page 5, ¶3). The Examiner stated "[providing an] efficient way to perform texture mapping process based on dimension type of texture data" as the motivation for combining Saunders, et al. with Lentz.

As stated above, the Examiner cited five (5) U.S. patents and one on-line computer dictionary in rejecting claim 1 as obvious under 35 U.S.C. §103. The Examiner applied the same reasoning to reject claim 21. However, the Examiner failed to distinguishably point out, in each

of five patents, the motivation and desirability to combine with features from the other four patents. The Examiner argued that all five cited references relate to texture processing and the motivation for combination would be to produce texture processing with time efficiency, properly retrieved texels and optimized hardware (OA, page 17, ¶2). The Examiner is accordingly making statements in hindsight with the guidance of the present application to combine the references. The Examiner's statement is without support from the cited references as required by MPEP 706.02(j) and *In Re Rouffet*.

Regarding claim 26, the Examiner admitted that Lentz does not disclose a texture packet having data relating to the dimensional type of its texture map, but the Examiner said that Saunders, et al. (U.S. Pat. No. 6,046,747) discloses such element (OA, page 14, ¶4). The Examiner further admitted that the combination of Lentz, Tanaka, et al., and Saunders, et al. does not disclose converting a multi-dimensional texture map into a one dimensional map, and the Examiner stated that Chimoto discloses this element (OA, page 15, ¶2). The Examiner cited four (4) patents to render claim 26 obvious and applied the same reason to claims 29 and 32 without pointing out where in these references the combination is suggested.

Regarding claim 9, the Examiner admitted that Lentz does not disclose a texture packet identifying the location of a texture map, and the Examiner stated that Tanaka, et al. discloses a packet data with storage location information (OA, page 8, ¶2). The Examiner further admitted that Lentz does not disclose a texture packet with data relating to the dimensional type of its texture map, and the Examiner, again, stated Saunders, et al. discloses such element (OA, page 8, ¶3). The Examiner cited three (3) patents to render claim 26 obvious and applied the same reason to claim 15 without pointing out where in these references the combination is suggested.

2. The Passages Cited by the Examiner Do Not Support the Examiner's Conclusions.

However, assuming, that arguendo, the Examiner's statements for claims 1 are admissible under MPEP 706.02(j) or *In Re Rouffet*, the passages cited by the Examiner do not support the Examiner's conclusion regarding the combination. In particular, the Examiner cited column 3, liens 51-54 in Kobayashi, et al. as providing motivation for multiple processing units accessing a single texture buffer, therefore combining Kobayashi, et al. with Lentz and Tanaka, et al. However, a closer and complete reading of the entire paragraph (col. 3, lines 49-60) reveals that

Kobayashi, et al. does not advocate using of a single texture buffer. Kobayashi, et al. states that "because a high speed, large capacity texture buffer is required, providing an equal number of texture buffers and parallel processing units greatly increases the scale of the apparatus for image generation" (col. 3, lines 54-57). Kobayashi, et al. further states that "using a single texture buffer requires a complex communications bus and an arbiter, thus prohibiting an unlimited increase in the number of parallel processing units." (col. 3, lines 57-60) Kobayashi, et al. actually teaches away from the invention by recommending against using a single texture buffer. Consequently, Kobayashi, et al. would not suggest combination of itself with Lentz and Tanaka, et al.

Further, the Examiner's statement on the motivation to incorporate the teaching of Tanaka, et al. in to the teaching of Lentz is not supported by the stated purpose of Lentz. The Examiner stated that the motivation to combine Lentz and Tanaka, et al. would be to provide enhanced image data by converting the existing file format into a new improved format (OA, page 5, ¶2). However, Lentz stated its objective is to maximize the data retrieval speed (col. 3, lines 21-22), and its approach can yield greater speeds by minimizing the occurrences of memory-page changes (col. 3, lines 32-34). Lentz is not concerned with, and does not address the providence of enhanced image data. Thus, there is not intended suggestion in Lentz to motivate combination with Tanaka, et al.

3. It is Impermissible to Use the Claimed Invention as the Template for Hindsight Reconstruction.

When making a determination under 35 U.S.C. §103, it is impermissible simply to engage in a hindsight reconstruction of the claimed invention, using the applicant's structure as a template and selecting elements from references to fill the gaps. *In Re Gorman*, 933 F.2d 982 at 987 (CAFC 1991). The references themselves must provide some teaching to render the invention obvious.

As shown above, the five cited references do not in themselves motivate combination in the manner suggested by the Examiner. Further, the Examiner has not correctly shown any motivation absent of the impetus of the present invention.

It is improper to us the claimed invention as the template to select elements from the cited references to fill the gaps of the Examiner's argument in accordance with to *In Re Gorman*.

CONCLUSION

For the reasons enumerated above, Applicant believes that the Examiner's conclusion was in error and requests that all claims be allowed.

No addition fees are believed due. However, the Commissioner is hereby authorized to charge any additional fees which may be required, including any necessary extensions of time, which are hereby requested, to Deposit Account No. 50-2666.

5/18/04
Date

Li K. Wang
Registration No. 44,393

Arnall Golden Gregory LLP
1201 West Peachtree Street
Suite 2800
Atlanta, GA 30309-3450

Tel. 404-873-8734
Fax 404-873-8735
E-Mail: li.wang@agg.com

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being placed in the U.S. Mail and addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date written below.

Lucille Golden-Blakey
Lucille Golden-Blakey

5/18/04
Date

Appendix

1. A graphics accelerator for processing a graphical image, the graphics accelerator comprising:

a single texture buffer for storing texture maps and data relating to the texture maps stored in the texture buffer; and

a plurality of texture processors that perform texturing operations on the graphical image, the plurality of the texture processors retrieving texture packets from the single texture buffer,

each texture processor including a fetching engine that retrieves the texture packets, each texture packet being stored in the texture buffer and being associated with a texture map that is different than the texture maps associated with any other texture packet in the texture buffer, each texture packet including data relating to the location of its associated texture map in the texture buffer and data relating to the dimensional type of that texture packet's associated texture map.

4. The graphics accelerator as defined by claim 1 wherein the dimensional type of each texture map is one of a one dimensional texture map, a two dimensional texture map, and a three dimensional texture map.

5. The graphics accelerator as defined by claim 1 wherein the texture processor further includes:

an input for receiving a texture message indicating that a texture map is to be utilized by the texture processor, the fetching engine responsively retrieving selected texture packets from the single texture buffer in response to receipt of the texture message.

6. The graphics accelerator as defined by claim 5 wherein the texture processor further includes:

a parsing engine for parsing a fetched texture packet and determining information relating to the texture map associated with the fetched texture packet.

7. The graphics accelerator as defined by claim 6 wherein the information relates to the location in the texture buffer of the texture map associated with the fetched texture packet.

8. The graphics accelerator as defined by claim 6 wherein the information relates to the number of dimensions of the texture map associated with the fetched texture packet.

9. A method of applying texture to a graphical image employing a graphics accelerator with a plurality of texture processors, the method comprising:

locating a texture packet identifying the location of a texture map in a single memory device, wherein the texture packet is associated with the texture map that is different than texture maps associated with other texture packets;

parsing the texture packet to determine the location of the texture map;

retrieving, based upon the determined location, the texture map from the single memory device; and

applying the texture map to the graphical image.

10. The method as defined by claim 9 wherein the texture packet is located by accessing a record identifying the location of the texture packet.

11. The method as defined by claim 9 wherein the single memory device is texture memory.

12. The method as defined by claim 9 wherein the texture packet is stored in the single memory device

13. The method as defined by claim 9 further comprising

reconstructing the texture map after it is retrieved from the single memory device.

14. The method as defined by claim 13 wherein the texture packet includes data relating to the dimensional type of the texture map, the texture map being reconstructed by parsing the

texture packet to determine the dimensional type of the texture map, the texture map being reconstructed based upon the determined dimensional type of the texture map.

15. A computer program product for use on a computer system with a plurality of texture processors for applying texture to a graphical image, the computer program product comprising a computer usable medium having computer readable program code thereon, the computer readable program code including:

program code for locating a texture packet identifying the location of a texture map in a single memory device, wherein the texture packet is associated with the texture map that is different than texture maps associated with other texture packets;

program code for parsing the texture packet to determine the location and the number of dimensions of the texture map;

program code for retrieving, based upon the determined location, the texture map from the memory device; and

program code for applying the texture map to the graphical image.

16. The computer program product as defined by claim 15 wherein the program code for locating includes program code for accessing a record identifying the location of the texture packet.

17. The computer program product as defined by claim 15 wherein the single memory device is texture memory.

18. The computer program product as defined by claim 15 wherein the texture packet is stored in the single memory device

19. The computer program product as defined by claim 15 further comprising:

program code for reconstructing the texture map after it is retrieved from the single memory device.

20. The computer program product as defined by claim 19 wherein the texture packet includes data relating to the dimensional type of the texture map, the program code for reconstructing comprising:

program code for parsing the texture packet to determine the dimensional type of the texture map, the texture map being reconstructed based upon the determined dimensional type of the texture map

21. A graphics accelerator for processing a graphical image, the graphics accelerator comprising:

a single texture buffer for storing texture maps and data relating to the texture maps stored in the texture buffer; and

a plurality of texture processors that performs texturing operations on the graphical image, the plurality of the texture processors retrieving texture packets from the single texture buffer, each texture processor including a fetching engine that retrieves texture packets, each texture packet being stored in the texture buffer and being associated with a texture map that is different than the texture maps associated with any other texture packet in the texture buffer, each texture packet including data relating to the dimensional type of its associated texture map.

22. The graphics accelerator as defined by claim 21 wherein each texture packet includes data relating to the location of its associated texture map in the single texture buffer.

24. The graphics accelerator as defined by claim 21 wherein the texture processor further includes:

an input for receiving a texture message indicating that a texture map is to be utilized by the texture processor, the fetching engine retrieving selected texture packets from the texture buffer in response to receipt of the texture message.

25. The graphics accelerator as defined by claim 24 wherein the texture processor further includes:

a parsing engine that parses a fetched texture packet and determines information relating to the texture map associated with the fetched texture packet.

26. A method of storing a texture map in a single linear texture memory of a graphics accelerator, the method comprising:

A. determining the dimension of the texture map;

B. converting the texture map to a one dimensional texture map if the dimension of the texture map is determined to be more than one dimensional, the one dimensional texture map having a first number of consecutive data blocks;

C. locating a second number of consecutive memory locations in the single texture memory, the first number being equal to the second number; and

D. storing the one dimensional texture map in the located memory locations in the single textured memory.

27. The method as defined by claim 26 wherein the texture map is two dimensional, step B comprising:

B1. defining a plurality of data blocks within the texture map; and

B2. assigning a sequence number to each of the data blocks, the sequence numbers being consecutive numbers.

28. The method as defined by claim 26 wherein step D comprises:

Dl. consecutively storing each consecutive data block of the one dimensional texture map in the located memory locations.

29. A graphics accelerator for processing graphical request code, the graphics accelerator comprising:

a single linear texture memory for storing texture maps;

a plurality of texture processors that applies textures to items to be displayed, the plurality of the texture processors retrieving texture packets from the single texture memory, each texture processor including a texture map converter that converts texture maps having dimensions greater than one dimensional to a one dimensional texture map, each dimensional texture map having a first number of consecutive data blocks, the texture processor further

including means for locating a second number of consecutive memory locations in the texture memory, the first number being equal to the second number; and

means for storing the one dimensional texture map in the located memory locations in the single texture memory.

30. The graphics accelerator as defined by claim 29 wherein the texture map converter comprises:

means for defining a plurality of data blocks within the texture map; and

means for assigning a sequence number to each of the data blocks, the sequence numbers being consecutive numbers.

31. The graphics accelerator as defined by claim 29 the storing means comprises:

means for consecutively storing each consecutive data block of the one dimensional texture map in the located memory locations.

32. A computer program product for use on a computer system for storing a texture map in a single linear texture memory of a graphics accelerator, the computer program product comprising a computer usable medium having computer readable program code thereon, the computer readable program code including

program code for determining the dimension of the texture map;

program code for converting the texture map to a one dimensional texture map if the dimension of the texture map is determined to be more than one dimensional, the one dimensional texture map having a first number of consecutive data blocks;

program code for locating a second number of consecutive memory locations in the texture memory, the first number being equal to the second number; an

program code for storing the one dimensional texture map in the located memory locations in the single texture memory.

33. The computer program product as defined by claim 32 wherein the texture map is two dimensional, the program code for converting comprising:

program code for defining a plurality of data blocks within the texture map; and

program code for assigning a sequence number to each of the data blocks, the sequence numbers being consecutive numbers.

34. The computer program product as defined by claim 32 wherein the program code for storing comprises

program code for consecutively storing each consecutive data block of the one dimensional texture map in the located memory locations.

35. A data structure for storing data relating to a texture map, the texture map having an associated dimension and being stored at a given location in a single memory device, the data structure comprising

a location field identifying the given location in the memory device; and

a dimension field identifying the dimension of the texture map.

36. The data structure as defined by claim 35 wherein the texture map comprises a set of mipmaps, further wherein the location field includes a plurality of subfields, each subfield identifying the location of one mipmap in the set of mipmaps.

37. The data structure as defined by claim 35 wherein the texture map spans a plurality of addresses in the single memory device, the location field identifying the plurality of addresses.

38. The data structure as defined by claim 35 wherein the data structure is stored in the single memory device, the single memory device being texture memory.